

GetPoint.txt

功能：计算NURBS曲线上参数u所对应的点(px, py)及其权因子w0。

输入参数：(m_xAVertex, m_yAVertex)-控制顶点； m_Weights-权因子； m_aNode-节点矢量，都是受保护成员。 m_nTimes - 曲线次数，属公有成员； u-参数值。

输出参数： px, py, w0-分别是NURBS曲线上参数u所对应的点的两个坐标与权因子。

调用函数：无

```
void GetPoint(double u, double &px, double &py, double &w0)
{
    CArray<double, double> ax, ay;
    CArray<double, double> aw;
    int k=m_nTimes;
    int i=0;
    for(int iN=m_aNode.GetSize()-k-2; iN>=0; iN--)
    {
        if(u>=m_aNode[iN])
        {
            i=iN;
            break;
        }
    }
    for(int j=i-k; j<=i; j++)
    {
        ax.Add(m_xAVertex[j]*m_Weights[j]);
        ay.Add(m_yAVertex[j]*m_Weights[j]);
        aw.Add(m_Weights[j]);
    }
    //德布尔算法
    for(int l=1; l<=k; l++)
    {
        for(j=i-k+l; j<=i; j++)
        {
            double alpha, du=m_aNode[j+k+1-l]-m_aNode[j];
            if(du==0.0) alpha=0.0;
            else alpha=(u-m_aNode[j])/du;
            int n=j-(i-k+1);
            ax[n]=(1-alpha)*ax[n]+alpha*ax[n+1];
            ay[n]=(1-alpha)*ay[n]+alpha*ay[n+1];
            aw[n]=(1-alpha)*aw[n]+alpha*aw[n+1];
        }
    }
    px=ax[0]/aw[0];
    py=ay[0]/aw[0];
    w0=aw[0];
    ax.RemoveAll();
    ay.RemoveAll();
    aw.RemoveAll();
}
```